J. Symbolic Computation (1987) **3**, 117–131

Path of Subterms Ordering and Recursive Decomposition Ordering Revisited

MICHAEL RUSINOWITCH

Centre de Recherche en Informatique de Nancy, Campus Scientifique BP 239, 54506 Vandœuvre Cedex, France

The relationship between several simplification orderings is investigated: the path of subterms ordering, the recursive path ordering and the recursive decomposition ordering. The recursive decomposition ordering is improved in order to deal with more pairs of terms, and is made more efficient and easier to handle, by removing useless computations.

Introduction

Rewriting systems enable one to prove equalities in systems described by equations. Their principle is based on orienting axioms (and some of their consequences called critical pairs) so that they are always applied in the same direction. Oriented equations are called *rewrite rules* and applying them is called *rewriting*. Each term is associated with a *normal form*, that is, a term that cannot undergo more rewriting. Equality between two terms is then equivalent to identity of their normal forms, provided the system is confluent. See, for example, Huet & Oppen (1980).

To be able to compute normal forms, rewritings must terminate. This is usually demonstrated by exhibiting a well-founded ordering that contains the rewriting relation. Dershowitz (1982) has shown that the simplification orderings are well founded.

In the following, we study four simplification orderings:

— the Path of Subterms Ordering (PSO) of Plaisted (1978);
— the Recursive Path Ordering (RPO) of Dershowitz (1979, 1982);
— the Recursive Decomposition Ordering (RDO) of Jouannaud *et al.* (1982);
— the Path Ordering (KNS) of Kapur *et al.* (1985).

All these orderings are defined by extending a basic ordering on function symbols, called a *precedence*.

In the first part of this work, we recall the definition of PSO and show that it contains RPO. Then, following the remarks of Dershowitz (1982, p. 293), we get, as a consequence, that PSO is well founded if and only if the precedence is well founded. A counterexample which shows that PSO is not included in RDO is given. By a slight change in the definition of PSO, we get a new ordering which possesses the following property called *incrementality* (see Jouannaud *et al.*, 1982): given two terms, the precedence can be automatically extended in order to orient them.

Our goal in the second part is to improve the implementation of RDO by eliminating many unnecessary comparisons. For example, comparing the terms $a(b(c))$ and $A(b(c))$ with the precedence $a < A$, generates four checks of "$a < A$". The idea is to first simplify

0747-7171/87/010117+15 $03.00/0

the terms by their common suffix, then compare the left parts of these terms: the result is now obtained with a single use of "$a < A$". The same remark is true for KNS. Based on this idea, a simplified version of RDO is proposed, one that does not require the fourth component of the elementary decompositions (called the "context" in Jouannaud *et al.*, 1982). We prove that the ordering thus obtained is equivalent to RDO. But this definition of RDO generates less computations than the previous ones. The section ends with an extension of RDO which is proven equivalent to KNS.

1. Preliminaries

1.1. MULTISET ORDERINGS

A *multiset* is a set with possibly repeated elements (see Dershowitz & Manna, 1979, or Jouannaud & Lescanne, 1982 for a full description). More formally, a multiset M is a mapping $E \to N$, where E is a set and N is the set of natural numbers. The set of all the multisets on E with finite carrier is denoted by $M(E)$. For x in E, we say that $M(x)$ is the number of occurrences of x in M, and we write $x \in M$ instead of $M(x) > 0$. The sum of two multisets M and N is the multiset $M+N$ such that $M+N(x) = M(x)+N(x)$ for all $x \in E$. More generally, if $M_1, M_2, \ldots, M_k$ are multisets,

$$\left(\sum_{s=1}^{k} M_s\right)(x) = \sum_{s=1}^{k} M_s(x).$$

Each ordering on E can be extended to $M(E)$:

DEFINITION 1.1. Let $<E$ be an ordering on E. We can define an ordering on $M(E)$ by:

$$M1 \ll E\ M2 \text{ iff } M1 \neq M2$$

and

$$\forall x \in E\ (M2(x) < M1(x) \Rightarrow \exists y \in E\ x < E\ y \quad \text{and} \quad M1(y) < M2(y)).$$

If we denote the minimum of two mappings $M1$ and $M2$ by $M1 \cap M2$, we get a generalisation of the intersection of two sets. The next lemma states that, to compare multisets, we may first delete their common instances:

LEMMA 1.2.

$$M1 \ll E\ M2 \textit{ iff } (M2 \neq \emptyset \textit{ and } \forall x \in M1-(M1 \cap M2)\ \exists y \in M2-(M1 \cap M2)\ x < E\ y).$$

Inclusion of orderings is preserved when they are extended to multisets:

LEMMA 1.3. *The multiset extension of an ordering is monotonic, that is to say:*

$$(\forall x, y \in E\ x < E\ y \Rightarrow x <' E\ y) \Rightarrow (\forall M1, M2 \in M(E)\ M1 \ll E\ M2 \Rightarrow M1 \ll' E\ M2).$$

We can use this property to prove that well foundedness is preserved, too, by the multiset extension. Our proof does not use Konig's lemma as others usually do.

LEMMA 1.4. *$\ll E$ is well founded iff $<E$ is well founded.*

PROOF. $<E$ can be extended to a total well-founded ordering $<'E$; now, $\ll' E$ is well founded because it can be seen as a lexicographical ordering on ordered words on E. Since $\ll E$ is included in $\ll' E$ by Lemma 1.3, it is also well founded.

1.2. TERMS—OCCURRENCES

In the following, F denotes a finite set of function symbols, and ar is the arity of symbols in F. X is a set of variables. For any set E, E^* is the set of all finite sequences of elements of E. The empty sequence is denoted by ε. Concatenation of sequences is indicated by a dot:

$$(x_1, \ldots, x_k).(y_1, \ldots, y_l) = (x_1, \ldots, x_k, y_1, \ldots, y_l).$$

Thus, N^* is the set of all finite sequences of positive integers. Let $T(F, X)$ be the set of terms, that is the set of functions:

$t: N^* \to FUX$ whose domain $occ(t)$ is finite and satisfies:

$$\begin{cases} \varepsilon \in occ(t) \\ u.i \in occ(t) \text{ iff } u \in occ(t) \text{ and } i \in [1, ar(t(u))]. \end{cases}$$

Let t/u be the subterm of t at occurrence u, $t(u)$ the function symbol at u in t and $|t| = card(occ(t))$ the size of t. $T(F)$ is the set of closed terms.

EXAMPLE. If $t = f(g(a), g(h(a, b, c)))$ and $u = 213$, then $t/u = c$.

1.3. SIMPLIFICATION ORDERINGS

Simplification orderings have been introduced by Dershowitz (1979, 1982).

DEFINITION 1.5. An ordering $<$ on $T(F, X)$ is a *simplification ordering* if it has the following properties for every function symbol f:

compatibility property: $t_1 < t_2 \Rightarrow f(\ldots, t_1, \ldots) < f(\ldots, t_2, \ldots)$

subterm property: $t < f(\ldots, t, \ldots)$.

Orienting rules from left to right according to a simplification ordering ensures that the rewriting process terminates:

THEOREM 1.6 (Dershowitz, 1979). *A rewriting system with finitely many symbols is terminating if there exists a simplification ordering $<$ such that for all substitutions s and for all rule $g \to d$, $s(d) < s(g)$.*

Now, we are going to study two particular simplification orderings: the path of subterms ordering (PSO) and the recursive decomposition ordering (RDO).

2. Path of Subterms Ordering

Rather than comparing two terms directly, PSO compares two data structures built up from these terms: their paths of subterms. A path of subterms is the sequence of subterms on a path from the root to a leaf.

2.1. PATHS OF SUBTERMS

DEFINITION 2.1. Path of subterms.

The multiset of paths of subterms of t is

$$\mathrm{SPATH}(t)=\begin{cases}\{t\} \text{ if } t \text{ is a constant or a variable}\\ \sum_{i=1}^{m}\{t\}.\mathrm{SPATH}(t_i) \text{ if } t=f(t_1,t_2,\ldots,t_m),\end{cases}$$

where

$$\{t\}.\mathrm{SPATH}(t_i)=\sum_{g\in\mathrm{SPATH}(t_i)}\{t.g\}.$$

We define also:

$$\mathrm{PSPATH}(t)=\begin{cases}\emptyset \text{ if } t \text{ is a constant or a variable}\\ \sum_{i=1}^{m}\mathrm{SPATH}(t_i) \text{ if } t=f(t_1,\ldots,t_m).\end{cases}$$

EXAMPLE:

$$t=f(a,a,g(c))$$
$$\mathrm{SPATH}(t)=\{(t,a),(t,a),(t,g(c),c)\}$$
$$\mathrm{PSPATH}(t)=\{(a),(a),(g(c),c)\}.$$

DEFINITION 2.2. Permutative congruence $\sim$.

$$f(s_1,\ldots,s_n)\sim g(t_1,\ldots,t_n) \text{ iff } f=g \quad\text{and}\quad s_i\sim t_{\pi(i)}$$

for some permutation π of $\{1,2,\ldots,n\}$.

DEFINITION 2.3. Let α be a sequence of terms, then SUBSEQU(α) is the multiset of subsequences of α

$$\mathrm{SUBSEQU}(\alpha)=\begin{cases}\emptyset \text{ if } \alpha \text{ is empty}\\ \{t,\varepsilon\}.\mathrm{SUBSEQU}(\beta)=t.\mathrm{SUBSEQU}(\beta)+\mathrm{SUBSEQU}(\beta) \text{ if } \alpha=t.\beta.\end{cases}$$

REMARK 2.4. If α is a path of subterms of t, then SUBSEQU(α) is actually a set, since every subterm of the path is distinct.

DEFINITION 2.5. If $<$ is an ordering on terms, then $<_{\mathrm{lex}}$ is a lexicographic-like ordering on sequences of terms defined by:

$$s<_{\mathrm{lex}}t \text{ iff } s=\emptyset \quad\text{and}\quad t\neq\emptyset$$

or

$$s_1<t_1$$

or

$$s_1\sim t_1 \quad\text{and}\quad (s_2,\ldots,s_m)<_{\mathrm{lex}}(t_2,\ldots,t_n)$$

where

$$s=s_1.s_2\ldots s_m \quad\text{and}\quad t=t_1.t_2\ldots t_n.$$

We are going to use the lexicographic extension of an intermediate ordering $<i$ on terms to get an ordering on paths of subterms, used in the definition of PSO.

2.2. THE PATH OF SUBTERMS ORDERING ON $T(F)$

DEFINITION 2.6. Let $<$ be a precedence on F, $<pso$ is defined recursively by:

$$s<pso\ t \text{ iff } \mathrm{SPATH}(s)\ll\mathrm{SPATH}(t),$$

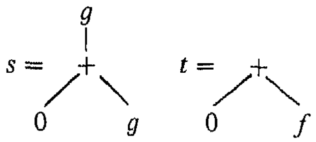
where <1 is an ordering on paths of subterms with:

$$a < 1\, b \text{ iff SUBSEQU}(a) \ll ilex \text{ SUBSEQU}(b),$$

where $<i$ is an ordering on terms with:

$u = f(u_1, \ldots, u_m) < i\, v = g(v_1, \ldots, v_n)$ iff $f < g$ or ($f = g$ and PSPATH$(u) \ll 1$ PSPATH(v)).

EXAMPLE 2.7.



with the precedence $0 < +$, $g < f$

SPATH$(s) = \{(s, 0+g, 0), (s, 0+g, g)\}$ SPATH$(t) = \{(t, 0), (t, f)\}(s, 0+g, 0) < 1(t, f)$ because $s < i\, f$, $0+g < i\, t$, $0 < i\, t$ $(s, 0+g, g) < 1\ (t, f)$ in a similar way and then $s < pso\, t$.

We can prove this exactly as in Plaisted (1978):

PROPOSITION 2.8. *$<pso$ is a simplification ordering.*

Moreover, $<pso$ satisfies a compatibility property stronger than the one which is required in the definition of a simplification ordering:

PROPOSITION 2.9. *If*

$$\{s_1, \ldots, s_n\} \ll pso\, \{t_1, \ldots, t_n\} \text{ then } f(s_1, \ldots, s_n) < pso\, f(t_1, \ldots, t_n)$$

$<pso$ is also monotonic with respect to the precedence:

PROPOSITION 2.10. *If $<$ is included (as a relation) in $<'$, then $<pso$ is included in $<'pso$.*

2.3. COMPARISON OF PSO AND RPO

Let us now recall the definition of RPO (cf. Dershowitz, 1979):

$$t = g(t_1, t_2, \ldots, t_n) < rpo\ s = f(s_1, s_2, \ldots, s_m)$$

iff

$$(rpo1)\ f = g \text{ and } \{t_1, t_2, \ldots, t_n\} \ll rpo\, \{s_1, s_2, \ldots, s_m\}$$

or

$$(rpo2)\ g < f \text{ and for all } t_i\text{: } t_i < rpo\ s$$

or

$$(rpo3)\ not(g \leqslant f) \text{ and for some } s_i\text{: } t \leqslant rpo\ s_i.$$

The main result of this section is:

THEOREM 2.11.

$$t < rpo\ s \Rightarrow t < pso\ s.$$

SKETCH OF THE PROOF. By induction on $|s| + |t|$. If $t < rpo\ s$ by:

*rpo*1: then $\{t_1, t_2, \ldots\} \ll rpo\ \{s_1, s_2, \ldots\}$, and by induction hypothesis $\{t_1, t_2, \ldots\} \ll pso\ \{s_1, s_2, \ldots\}$; we can conclude $t < pso\ s$ from $f = g$ and the generalised compatibility property of $<pso$.

*rpo*2: then $g < f$ and for all t_i, $t_i <rpo\ s$. We have $\mathrm{SPATH}(s) \cap \mathrm{SPATH}(t_i) = \emptyset$, since $s \neq ti$. So, by induction, for each $a \in \mathrm{SPATH}(t_i)$, there exists $b \in \mathrm{SPATH}(s)$ with $a <1\ b$. Note that $t.a <1\ b$, by the following line of reasoning: $t.\mathrm{SUBSEQU}(a) \ll ilex\ (s)$ since $t <i\ s$ due to $g < f$; $\mathrm{SUBSEQU}(a) \ll ilex\ \mathrm{SUBSEQU}(b)$ because $a <1\ b$; so,

$$\mathrm{SUBSEQU}(t.a) = t.\mathrm{SUBSEQU}(a) + \mathrm{SUBSEQU}(a) \ll ilex\ \mathrm{SUBSEQU}(b)$$

because (s) belongs to $\mathrm{SUBSEQU}(b)$, but not to $\mathrm{SUBSEQU}(t.a)$.

From this we get $\mathrm{SPATH}(t) \ll 1\ \mathrm{SPATH}(s)$ and so, $t <pso\ s$.

*rpo*3: then $t \leqslant rpo\ s_i$; by induction $t \leqslant pso\ s_i$; we conclude from the subterm property and the transitivity of $<pso$.

Corollary 2.12. *If the precedence $<$ is total, then $<rpo$ and $<pso$ are the same ordering.*

Proof. If $<$ is total, then $<rpo$ is total on $T(F)/\sim$, and so is $<pso$ from the last theorem. And, if $s \sim t$, s and t are neither comparable by $<rpo$ nor by $<pso$.

Corollary 2.13. *If $<$ is well founded, then $<pso$ is well founded.*

Proof. $<$ is included in a total well-founded ordering $<'$. It is known that $<'rpo$ is well founded (cf. Dershowitz, 1979); but $<'rpo = <'pso$ and $<pso$ is included in $<'pso$ from the monotonicity property of PSO; the corollary follows as in Lemma 1.4.

Remark 2.14. It will be shown in the next section that PSO is not included in RDO.

2.4. A variant of PSO

In most cases, the use of subsequences for comparing paths of subterms yields redundant computations. Considering paths of subterms as multisets provides a simple variant of PSO, with an additional property of incrementality.

Definition 2.15. Let $<$ be a precedence. We define $<ps$ recursively:

$$s <ps\ t \text{ iff } \mathrm{SPATH}(s) \ll 2\ \mathrm{SPATH}(t),$$

where $a <2\ b$ iff $\alpha \ll j\ \beta$ (α and β denotes the multiset of subterms occurring in the path a and b, respectively) and

$$u <j\ v \text{ iff } (f < g) \quad \text{or } (f = g \text{ and } \mathrm{PSPATH}(u) \ll 2\ \mathrm{PSPATH}(v))$$

where $u = f(u_1, \ldots, u_n)$ and $v\,g(v_1, \ldots, v_m)$.

It can be proved that $<ps$ is also a simplification ordering, which is monotonic, and contains RPO. The next example shows that $<ps$ is also easy to use. Moreover, the precedence required to orient a rule can be easily computed. We conjecture that $<ps$ and PSO are the same ordering.

Example 2.16. Let us consider the rewriting system:

(1) $\neg(\neg(x)) \to x$

(2) $\neg(x \vee y) \to \neg(x) \wedge \neg(y)$

(3) $$\neg(x \wedge y) \rightarrow \neg(x) \vee \neg(y)$$
(4) $$x \wedge (y \vee z) \rightarrow (x \wedge y) \vee (x \wedge z)$$
(5) $$(y \vee z) \wedge x \rightarrow (y \wedge x) \vee (z \wedge x).$$

An empty precedence is enough to orient (1), because of the subterm property of $<ps$. To orient (2) we need to have:

$$\{\neg(x) \wedge \neg(y), \neg(x), x\} \ll j \{\neg(x \vee y), x \vee y, x\},$$

but $\neg(x) <j \neg(x \vee y)$.

Hence (2) is oriented with $<ps$ iff $(\wedge < \neg)$ or $(\wedge < \vee)$. By exchanging the symbols '$\wedge$' and '$\vee$' we get the condition for (3): $(\vee < \neg)$ or $(\vee < \wedge)$. In order to have (4) directed with $<ps$, we need to have:

$$\{(x \wedge y) \vee (x \wedge z), x \wedge y, x\} \ll j \{x \wedge (y \vee z), x\},$$

but $(x \wedge y) <j (x \wedge (y \vee z))$. To bound $(x \wedge y) \vee (x \wedge z)$ for $<j$, the necessary and sufficient condition is $\vee < \wedge$. Then the other paths of the right-hand side of (4) are bounded too. Let us finally summarise the conditions $\vee < \wedge \sim \neg$.

3. RDO Revisited

3.1. Definition of RDO

A full description of RDO can be found in Jouannaud *et al.* (1982) or Lescanne (1982). The set $T(F, X, \square)$ contains terms with at most one terminal occurrence of $\square$, where $\square$ is a symbol not in F that can be viewed as the empty term. If X is empty, we denote this set as $T(F, \square)$. If u and v belong to N^*, then u/v is the word $w \in N^*$ such that $v.w = u$. Let t/u be the subterm of t at occurrence u, and $t[u \leftarrow t']$ the term obtained by replacing t/u by t' in t. If t belongs to $T(F, X, \square)$

$$\|t\| = \|\{u \in occ(t); t(u) \neq \square \text{ and } t(u) \notin X\}\|.$$

A *path* p of a term t is an occurrence such that $ar(t(p)) = 0$. Let p be a path of t, u a strict prefix of p, i the integer such $u.i$ is a prefix of p; $u.i$ is denoted by $suc(u, p)$.

Definition 3.1. Elementary decomposition.

Given $t \in T(F, \square)$, p, a path of t and u, a prefix of p, the *elementary decomposition* $d_u^p(t)$ *of t in u along the path p* is:

$\emptyset$ if $t(u) = \square$; otherwise it is the quadruple: $\langle g, a, \psi, C\rangle$, where

$$g = t(u)$$
$$a = d^{p/suc(u,p)}(t/suc(u.p))$$

ψ is the multiset of other subterms of t/u, that is:

$$\{t/u.j : 1 \leqslant j \leqslant ar(t(u)) \text{ and } u.j \neq suc(u, p)\}$$
$$C = d^u(t[u \leftarrow \square])\rangle,$$

where $d^p(t)$ is the *decomposition of t along the path p*, that is the set

$$\{d_u^p(t) : u \text{ is a prefix of } p\}.$$

We define also the multiset $d(t) = \{d^p(t) : p \text{ is a path of } t\}$.

REMARK. The term $t[u \leftarrow \square]$ is called the *context* of u in t. ψ is called the *neighbouring part*.

DEFINITION 3.2. RDO.

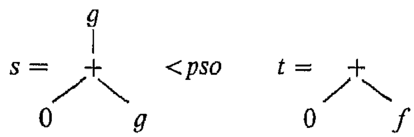
Given a partial ordering on F, we define the *recursive decomposition ordering* in the following way: $s <rdo\ t$ iff $d(s) \ll* \ll* d(t)$, where $\ll* \ll*$ stands for the multiset of multisets ordering extending $<*$, and $<*$ is defined in the following way:

$$d_u^p(s) = \langle f, a, \phi, c \rangle <* d_v^q(t) = \langle g, b, \psi, d \rangle$$

iff one of the following holds:

*dec*1: $f < g$
*dec*2: $f = g$ and $a \ll* b$
*dec*3: $f = g$ and $a = b$ and $\phi \ll rdo\ \psi$
*dec*4: $f = g$ and $a = b$ and $\phi = \psi$ and $c \ll* d$.

REMARK. PSO is not included in RDO. As shown in a previous example:



with $0 < +$ and $g < f$,

whereas s and t are incomparable under the definition of $<rpo$.

But we do not have $s <rdo\ t$ because it is impossible to bound $d^{11}(s)$ with a decomposition along a path of t. A consequence is that PSO strictly contains RPO. The following example, given in Jouannaud *et al.* (1982) shows that RDO is not included in PSO:

We take an empty precedence on $F = \{f, a, b\}$,

$$s = f(f(a, b), f(a, b))$$

and

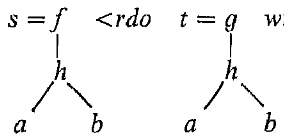
$$t = f(f(f(a, a), f(a, a)), f(f(b, b), f(b, b))).$$

It is not possible to compare s and t with PSO; nevertheless, we have $s <rdo\ t$.

However, it is possible to prove that PSO, RPO and RDO are the same ordering when restricted to monadic terms.

3.2. REMOVING CONTEXTS FROM THE DEFINITION OF RDO

EXAMPLE. Suppose we want to show:



with the precedence $f < g$

To get $d^{11}(s) \ll * \, d^{11}(t)$ we have to prove:

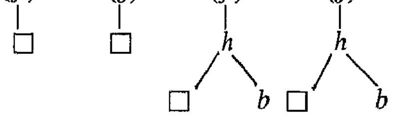
$$d^{11}_{11}(s) <* \, d^{11}_{11}(t) \text{ with } dec4$$

and

$$d^{11}_{1}(s) <* \, d^{11}_{1}(t) \text{ with } dec4.$$

But this computation is useless, because the last inequality is, in fact, a consequence of the previous one. Indeed:

$$d^1(f) \ll * \, d^1(g) \Rightarrow d^{11}(f) \ll * \, d^{11}(g)$$



In the following, we give a new (equivalent) definition of RDO that gets rid of this redundancy, by eliminating all contexts from decompositions. Clearly, terms can be reconstructed from their decompositions with contexts, but they can be reconstructed without contexts as well. As a consequence, contexts are not really needed, as we show next.

DEFINITION OF RD.

DEFINITION 3.3. Simple decomposition.

Given $t \in T(F)$, p, a path of t and u, a prefix of p, the *simple decomposition* $D^p_u(t)$ *of* t *in* u *along the path* p is the triple $\langle g, a, \psi \rangle$, where

$$g = t(u)$$
$$a = D^{p/suc(u,p)}(t/suc(u, p))$$
$$\psi \text{ is the multiset } \{t/u.j\colon 1 \leqslant j \leqslant ar(t(u)),\, u.j \neq suc(u, p)\}$$

and the *simple decomposition of* t *along the path* p is the set

$$D^p(t) = \{D^p_u(t)\colon \text{is prefix of } p\}.$$

Let us also define the multiset: $D(t) = \{D^p(t)\colon p \text{ is path of } t\}$.

DEFINITION 3.4. RD.

The *simplified recursive decomposition ordering RD* is defined as follows:

$$s < rd \; t \text{ iff } D(s) \ll \bigcirc \ll \bigcirc D(t)$$

with

$$D^p_u(s) = \langle f, a, \phi \rangle < \bigcirc \quad D^q_v(t) = \langle g, b, \psi \rangle$$

iff one of the following holds:

DEC1: $f < g$
DEC2: $f = g$ and $a \ll \bigcirc b$
DEC3: $f = g$ and $a = b$ and $\phi \ll rd \; \psi$.

REMARK 3.5. Note that we use a small d to denote elementary decompositions, while a capital D is used to denote simple decompositions.

RDO AND RD ARE EQUIVALENT.

If we need the contexts when comparing two paths p and q, using RDO, this implies that the two paths end with the same subsequence of subterms. Therefore, the last simple decompositions encountered on p are equal to the last simple decompositions of q. Hence, we do not need them when comparing p and q with RD, unlike RDO. This is the key aspect of our proof that RDO = RD. We state the previous remark more formally in the next lemma.

LEMMA 3.6. *Let p be a path of s, and q a path of t. Suppose that $D_u^p(s) = D_v^q(t)$, where u and v are prefixes of p and q, prespectively. Then:*

$$D^p(s) \ll\bigcirc\, D^q(t) \;\textit{ iff }\; D^u(s[u\leftarrow\Box]) \ll\bigcirc\, D^v(t[v\leftarrow\Box]).$$

We can now state the main result.

THEOREM 3.7. *Given p and q, two paths of s and t respectively, we have:*

$$d^p(s) \ll* d^q(t) \;\textit{ iff }\; D^p(s) \ll\bigcirc\, D^q(t).$$

Sketch of the proof. by induction on $\|s\|+\|t\|$.

First case: the paths p and q end with the same subsequence of subterms. So we can simplify p and q by their common suffix and are brought back to compare two paths u and v of $s[u\leftarrow\Box]$ and $t[v\leftarrow\Box]$, respectively, with u and v prefix of p and q, respectively. The result follows from the induction hypothesis.

Second case: p and q do not have the same tail. Hence, we never need contexts when comparing the elementary decompositions of p with those of q. Therefore, we could just as well perform the comparison with simple decompositions.

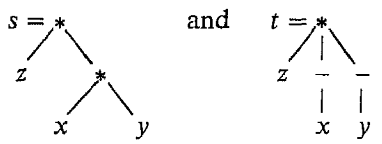
The last theorem yields immediately.

COROLLARY 3.8. *$s <rdo\ t$ iff $s <rd\ t$.*

When comparing two terms with RDO, we just need to compare the maximal decompositions of each term; this is the called $++$*strategy* in Jouannaud *et al.* (1982). This improvement is not possible with RD. However, Pierre Lescanne, who implemented both RD and RDO in his REVE system, noticed that, even with the $++$strategy, RDO appeared to be less efficient than RD, in most cases.

3.3. IMPROVING RDO

EXAMPLE 3.9. Given the terms:



with the precedence $* < -$, we can show that $s <rdo\ t$ is false. It is sufficient to prove that we have not $d^1(s) \ll* d^1(t)$. But

$$d^1(s) = \{\langle *, z, \{\underset{x\quad y}{*}\}, \Box\rangle\}, \qquad d^1(t) = \{\langle *, z, \{\underset{x}{-}, \underset{y}{-}\}, \Box\rangle\}$$

and

$$\begin{array}{c}\{*\}\\ \diagup\ \diagdown\\ x\qquad y\end{array} \ll rdo \begin{array}{c}\{-,-\}\\ \mid\ \ \mid\\ x\ \ y\end{array} \text{ is false.}$$

However, we have:

$$\begin{array}{c}d(*)\\ \diagup\ \diagdown\\ x\qquad y\end{array} \ll* \ll* \begin{array}{c}d(-)\\ \mid\\ x\end{array} + \begin{array}{c}d(-).\\ \mid\\ y\end{array}$$

On the other hand, we could verify: $s <pso\, t$ and $s <ps\, t$, so we have another counter-example showing that PSO is not included in RDO.

RDO fails to order s and t because it requires paths to be gathered in the neighbouring part of decompositions, so neither $-x$, nor $-y$ can bound $x*y$ and they cannot help each other to do that. More generally, if ϕ and ψ are two multisets of terms, then:

$$\phi \ll rdo\, \psi \text{ implies } \sum_{r\in\phi} d(r) \ll* \ll* \sum_{q\in\psi} d(q)$$

but the converse is false. So, if the last condition is taken instead of *dec*3 in the definition of RDO, the comparison should be more successful.

The previous example suggested to us an easy way to improve RDO. Instead of comparing the multisets of subterms that constitute the third part of decompositions, we compare the multiset sums of the paths of these subterms, without taking into account the original subterm they belong to.

In the following we shall extend RD in that direction. However, the same could be done with RDO itself.

DEFINITION 3.10. IRD.

We define on $T(F)$ the ordering *IRD* in the following way:

$$s <ird\, t \text{ iff } D(s) \ll\bullet \ll\bullet D(t),$$

where

$$D_u^p(s) = \langle f, a, \phi\rangle <\bullet D_v^q(t) = \langle g, b, \psi\rangle$$

iff one of the following holds:

DEC1: $f < g$
DEC2: $f = g$ and $a \ll\bullet b$
DEC3*b*: $f = g$ and $a = b$ and $\sum_{r\in\phi} D(r) \ll\bullet \ll\bullet \sum_{q\in\psi} D(q)$.

Now, with this slight change in the definition of RD, we get in Example 3.9: $s <ird\, t$, and we can prove straightforward that RDO is strictly included in IRD:

THEOREM 3.11. *$s <rdo\, t \Rightarrow s <ird\, t$.*

In Kapur *et al.* (1985), an ordering which we call KNS is described, and the authors give the following example of two terms that can be compared with their ordering, but for which RDO does not apply. This example is quite similar to Example 3.9.:

$$s = h(a(z), g(a(a(x)), x), g(b(b(y)), y))$$
$$t = h(a(z), g(a(x), b(y)), g(a(x), b(y))$$

and we may prove that $t <ird\, s$.

As a matter of fact, it will be shown in the next section that IRD and KNS are the same ordering. Now, if we compute the complexity of comparing two terms s and t using the definition of IRD, by the method described in Kapur *et al.* (1985), we get an upper bound $O(|s|^4 * |t|^4)$. With KNS, the upper bound is $O(|s|^5 * |t|^5)$. This is not surprising, since this last ordering uses contexts.

3.4. IRD AND KNS ARE EQUIVALENT

We proceed as follows:

(A) We give an alternative definition of KNS, which we think is simpler and more efficient.
(B) We show that IRD and KNS are equivalent.

(A) ALTERNATIVE DEFINITION OF KNS

In this subsection, the notations and definitions are taken from Kapur *et al.* (1985). We suppose, for simplicity, that no variable is involved.

DEFINITION 3.12. A *K-path* is a sequence of two-tuples, with the following properties:

Let $P = \langle f_1, T_1\rangle \dots \langle f_n, T_n\rangle$ be a K-path. Then

1. f_i is the top level symbol of T_i for $1 \leqslant i \leqslant n$.
2. T_{i+1} is an immediate subterm of T_i.

With every couple $\langle f_i, T_i\rangle$ in the K-path P we can associate a *left-context* (LC) and a *right-context* (RC) defined as:

$$\mathrm{RC}(\langle f_i, T_i\rangle, P) = \begin{cases} \varepsilon \text{ if } i = n \\ \langle f_{i+1}, T_{i+1}\rangle \dots \langle f_n, T_n\rangle \text{ if } i < n \end{cases}$$

$$\mathrm{LC}(\langle f_i, T_i\rangle, P) = \begin{cases} \varepsilon \text{ if } i = 1 \\ \langle f_1, T_1\rangle \dots \langle f_{i-1}, T_{i-1}\rangle \text{ if } i > 1. \end{cases}$$

The K-path P is a *full K-path of a term t* if $T_1 = t$ and T_n is a constant. The multiset of the full K-paths of a term t is denoted by MP(t). The KNS ordering is defined by $s <kns\ t$ if $\mathrm{MP}(s) \ll p\ \mathrm{MP}(t)$, where $<p$ is an ordering on the K-paths and is defined above:

Let $P1 = \langle k_1, T_1\rangle . \langle k_2, T_2\rangle \dots \langle k_m, T_m\rangle$ and $P2 = \langle h_1, S_1\rangle \dots \langle h_n, S_n\rangle$ be two K-paths.

We shall say that $P1 = P2$ if $m = n$ and $k_i = h_i$ and $T_i \sim S_i$ for all $i \in \{1, \dots, n\}$. The K-path comparison is performed as follows:

$P2 <p\ P1$ iff for all $\langle h_j, S_j\rangle$ in $P2$ there is $\langle k_i, T_i\rangle$ in $P1$ such that:

a. $h_j < k_i$, or
b. $h_j = k_i$ and
   1. $\mathrm{RC}(\langle h_j, S_j\rangle, P2) <p\ \mathrm{RC}(\langle k_i, T_i\rangle, P1)$ or
   2. $\mathrm{RC}(\langle h_j, S_j\rangle, P2) = \mathrm{RC}(\langle k_i, T_i\rangle, P1)$ and $S_j <kns\ T_i$ or
   3. $\mathrm{RC}(\langle h_j, S_j\rangle, P2) = \mathrm{RC}(\langle k_i, T_i\rangle, P1)$ and $S_j = T_i$ and $\mathrm{LC}(\langle h_j, S_j\rangle, P2) <p\ \mathrm{LC}(\langle k_i, T_i\rangle, P1)$.

Now let us have a closer look at condition 3. This condition plays the role of the "context comparison" in RDO. If $P1$ and $P2$ have no common suffix, then we never have to use the test b.3. We can always reduce the situation to the previous one, thanks to the next result, which states a converse of Lemma 5 of Kapur *et al.* (1985).

LEMMA 3.13.: *Let P1, P2, P3, P4, P5 be K-paths such that* $P4 = P1 . P3$ *and* $P5 = P2 . P3$. *Then* $P5 <p\, P4$ *iff* $P2 <p\, P1$.

This leads to a more efficient definition of $<p$ and the KNS ordering.

PROPOSITION 3.14. *Let* $P1'$ *and* $P2'$ *be the K-paths we get from P1 and P2 by deleting their common suffix. Then* $P2 <p\, P1$ *iff for all* $\langle h_j, S_j \rangle$ *in* $P2'$ *there exists* $\langle k_i, T_i \rangle$ *in* $P1'$ *such that a. or b.1. or b.2. is true.*

We are now ready to prove:

(B) IRD AND KNS ARE EQUIVALENT

It is sufficient to prove that the path comparisons using $\ll\bullet$ or $<p$ always yield the same result.

PROPOSITION 3.15. *Let P and Q be two full K-paths of s and t, respectively. Then* $P <p\, Q$ *iff* $D^p(s) \ll\bullet\, D^q(t)$, *where* $p = u_1 \ldots u_n$ *and* $q = v_1 \ldots v_m$ *are paths of s and t, respectively, that verify:*

$$P = \langle s(\varepsilon), s \rangle . \langle s(u_1), s/u_1 \rangle \ldots \langle s(p), s/p \rangle$$

and

$$Q = \langle t(\varepsilon), t \rangle . \langle t(v_1), t/v_1 \rangle \ldots \langle t(q), s/q \rangle.$$

PROOF. (By induction on $|s| + |t|$) assume $P <p\, Q$, and let P' and Q' be the K-paths we get by deleting the common suffix of P and Q. Let $\langle h_j, S_j \rangle$ be in P', with u such that $S_j = s/u$. We suppose that the $\langle k_i, T_i \rangle$ in Q' takes care of $\langle h_j, S_j \rangle$, and v is such that $T_i = t/v$. Let us show that $D^p_u(s) <\bullet\, D^q_v(t)$. The only non-trivial case to consider is b.3, that is, $h_j = k_i$ and

$$\mathrm{RC}(\langle h_j, S_j \rangle, P) = \mathrm{RC}(\langle k_i, T_i \rangle, Q)$$

and $S_j <kns\, T_i$. We have by definition of $<kns$: $\mathrm{MP}(S_j) \ll p\, \mathrm{MP}(T_i)$. The relevant K-paths of this last inequality belong to smaller terms. Therefore, we can apply the induction hypothesis:

$$\sum_{a \in \mathrm{MP}(S_j)} D(a) \ll\bullet \ll\bullet \sum_{b \in \mathrm{MP}(T_i)} D(b).$$

The relation above is nothing else but DEC3b.

A similar argument proves the converse.

3.5. INCORPORATING STATUS IN RD

In the ordering RDOS defined by Lescanne (1984), when comparing two elementary decompositions having the same "leading symbol", the *status* of this symbol gives information about how to go on with the comparison: by comparing the "neighbouring part" as multisets or as lexicographically ordered sequences. The lexicographic statuses are very useful to orient associativity laws. They were first introduced by Kamin & Levy (1980) in order to extend RPO.

We suggest generalising the notion of status to include more strategies than the two, multiset and lexicographic. We may, for instance, assign a "depth-first" or "breadth-first" status to some symbols according to how we handle the components of decompositions.

First, we modify slightly the definition of simple decompositions.

DEFINITION 3.16. Given p a path of t and u an occurrence of p, we define $D_u^p(t)$ as the triple $\langle f, a, \psi \rangle$, where

$$f = t(u)$$
$$a = D^{p/suc(u,p)}$$
$$\psi \text{ is the } sequence\ \langle t/u.j : 1 \leqslant j \leqslant ar(f) \rangle,$$

where $D^p(t)$ and $D(t)$ are defined as usual.

DEFINITION 3.17. Given an ordering $<x$ on the elementary decompositions, the recursive decomposition ordering with status is defined in the following way:

$$s <rds\ t \text{ iff } D(s) \ll x \ll x D(t).$$

The only property required for $<x$ is:

$$t_i <rds\ t'_i \text{ implies } \langle f, a, (t_1, \ldots, t_i, \ldots, t_n) \rangle <x \langle f, a, (t_1, \ldots, t'_i, \ldots, t_m) \rangle.$$

Then, without any other condition, we can prove that $<rds$ is a simplification ordering (see Kamin & Levy, 1980). Now let us give a good candidate for $<x$:

DEFINITION 3.18.

$$D_u^p = \langle f, a, \phi \rangle <x\ D_v^q = \langle g, b, \psi \rangle \text{ iff } f < g \text{ or } f = g$$

and case status of f is:

multiset-depth-first: (RDO-like)	$a \ll x\, b$ or $a = b$ and $\sum_{S \in \phi} d(S) \ll x \ll x \sum_{T \in \psi} d(T)$
multiset-breadth-first:	$\sum_{S \in \phi} d(S) \ll x \ll x \sum_{T \in \psi} d(T)$
lexicographical-*lr*: (KAMIN&LEVY-like)	$\phi <lr\ \psi$ where $<lr$ is the left to right lexicographic extension of *rds* to sequences of term

and so on. . . .

We can add other cases as long as the property in Definition 3.17 is satisfied. By varying the choices of status for function symbols, RDS can take on features of the previously studied simplification orderings.

PROPOSITION 3.19.

1. *If all the statuses are multiset-depth-first, then* RDS = KNS = IRD, *but the time complexity of comparing s and t with* RDS *has an upper bound* $O(|s|^4 * |t|^4)$ *instead of* $O(|s|^5 * |t|^5)$.
2. *If the statuses are multiset depth first or lexicographic, then* RDS *properly contains* RPOS *and* RDOS, *respectively.*

REMARK. We do not know of any example that PSO can handle, but RDS cannot.

So, RDS is made more practical than others by its ability to include semantics for each function symbol by a generalised notion of status.

Conclusion

PSO and RDO have many common features: both of them work on paths of subterms, and extend RPO. Furthermore, PSO can be easily expressed in terms of decompositions. They essentially differ when comparing subterms with the same roots: PSO splits the paths and compares, in parallel, all paths of subterms issuing from the root; on the other hand, RDO goes on with the same path and checks the other paths later on. In other words, PSO works breadth first, and RDO works depth first. This is why these two orderings do not always give the same result. But we have been able to improve RDO by incorporating some ideas of PSO. We can sum up our results in the following diagram:

$$\mathrm{PSO} \supset \underset{\underset{\mathrm{PS}}{\cap}}{\mathrm{RPO}} \subset \mathrm{RDO} = \mathrm{RD} \subset \mathrm{IRD} = \mathrm{KNS}$$

An interesting feature of RD is its conceptual simplicity, which makes modifications easier. In particular, we can incorporate "status" à la Kamin & Levy (1980) in RD by comparing decompositions in a lexicographic way (instead of using multisets). Besides symbols with lexicographical status, one may have symbols with "depth-first" status or "breadth-first" status. So, according to the choice of the status of the function symbols, one can make RD more or less similar to either RDO or PSO.

I would like to thank Pierre Lescanne for his helpful guidance, and Jieh Hsiang, Jean-Pierre Jouannaud, Emmanuel Kounalis, Alain Quere and Jean-Luc Remy for reading the manuscript.

References

Dershowitz, N. (1979). *Orderings for Term Rewriting Systems.* Proc. 20th Symposium on Foundations of Computer Science.

Dershowitz, N. (1982). Orderings for term-rewriting systems. *Theor. Comp. Sci.* **17**, 279–301.

Dershowitz, N., Manna, Z. (1979). Proving termination with multiset orderings. *Commun. ACM* **22**, 465–476.

Huet, G., Oppen, D. C. (1980). Equations and rewrite rules: a survey. In: (Book, R., ed.) *Formal Language Theory: Perspectives and Open Problems*, pp. 349–405. London: Academic Press.

Jouannaud, J.-P., Lescanne, P., Reinig, F. (1982). Recursive decomposition ordering. In: (Bjormer, D., ed.) *Formal Description of Programming Concepts*, pp. 331–348. Amsterdam: North-Holland.

Jouannaud, J.-P., Lescanne, P. (1982). On multiset orderings. *Inf. Proc. Lett.* **15**, 57–63.

Kamin, S., Levy, J.-J. (1980). *Two Generalizations of the Recursive Path Orderings.* Unpublished note. Department of Computer Science, University of Illinois, Urbana, IL.

Kapur, D., Narendran, P., Sivakumar, G. (1985). A path ordering for proving termination of term rewriting systems. *CAAP 85.*

Lescanne, P. (1982). Some properties of decomposition orderings, a simplification ordering to prove termination of rewriting systems. *RAIRO Information theorique* **4**, 331–347.

Lescanne, P. (1984). Uniform termination of term rewriting systems—Recursive decomposition ordering with status. *CAAP 84.*

Plaisted, D. (1978). *A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems.* University of Illinois, Sept. 78, UIUCDCS-R-78-943.